A Novel Hybrid Algorithm of PCA and SPIHT methods in Medical Image Compression

Chandrashekhar Kamargaonkar
F.E.T, SSTC, SSGI,
Junwani, Bhilai, India
amol_kamar@rediffmail.com

Monisha Sharma
F.E.T., SSTC, SSGI,
Junwani, Bhilai, India
monisha.sharma10@gmail.com

***Abstract*—Compression of medical picture has acquired great attention attributable to its raising need to decrease the picture size while not compromising the diagnostically crucial medical data exhibited on the picture. PCA algorithm may be used to help in picture compression. In this paper, a comparative study is provided for PCA and SPIHT compression method. Here PCA algorithm is characterized in two forms i.e. Standard PCA and Block-Based PCA. The block based PCA has 2 extended-PCA algorithms that manipulate the block data of the picture are evaluated. The 1st algorithm is referred to as block-by-block PCA where standard PCA algorithm is utilized on every block of the picture. In the next algorithm- the block-to-row PCA, all of block data are initially concatenated into a row before the standard PCA algorithm is therefore utilized in the remodelled matrix. In this work, the SPIHT is being compared with the above two methods in terms of image quality and compression ratio. With this work, it's observed that block-based PCA performs superior to the PCA algorithm and SPIHT with regards to picture quality, producing similar compression ratio like the PCA algorithm.**

Keywords—Medical Image Compression; Principal Component Analysis (PCA); Block-Based PCA; Compression ratio; Image quality; SPIHT

I. INTRODUCTION

Clinical imaging has had an excellent impact on the identification of diseases and surgical planning. However, imaging instruments still generate a lot of information per patient, usually one thousand images or ~500 MB. These information requires large storage and economical transmission [20]. Regardless of larger improvement in transmission storage space and communication technologies, the clinical image compression plays the demanding role. In telemedicine, medical pictures generated from clinical centers with efficient image acquisition contraptions equivalent to for example Computerized Tomography (CT), Resonance Imaging (MRI), Ultrasound (US), Electrocardiogram (ECG) and Positron Emission Tomography (PET) have to be compelled to be transmitted handily over the network for studying by another medical professional. The massive house is going to be occupied by this picture and so it cost high and also the communication gets affected attributable to high traffic throughout transmission. For this rationale, there may be a desire for medical compression so as to cut back the storage and bandwidth needs. With the exception of conserving essential info in the medical picture, high compression ratio and capability to decode the compressed pictures at various qualities will be the major concerns in medical picture compression [4].

Principal Component Analysis (PCA) is normally a statistical way that linearly remodels an original group of variables into smaller group of uncorrelated variables that signify the largest variance from the multivariate input data. PCA has been widely utilized in face recognition considering it is employed to extract the fundamental feature of the high dimensional data space [4]. It's additionally got growing attention in picture compression as its application supported by the theory that the 'important information' is captured with the aid of the primary component. In picture compression, PCA can be referred to as Karhunen-Loeve (KL) Transform or Hotelling Transform wherever the computation of the variance - covariance matrix of the info is diagnosed using Singular Value Decomposition (SVD).

Principal Component Analysis (PCA) is usually a lossy compression scheme that achieves picture compression by remodelling a sophisticated data set to a lesser dimension [5]. Lossy compression can lead to loss of data and this method cannot be reversed. However, lossy compression remains a stimulating analysis space attributable to its comparative higher compression ratios. In most applications, the tiny degree of error due to lossy compression is tolerable, as an instance, conferencing of video. We've to be cautious once lossy compression is utilized on medical pictures, mainly because coding error might the diagnostic values or maybe ends up in inaccurate diagnoses. At the same time, lossy compression offers associate economical means for remote diagnoses and safe-keeping of medical pictures. It's crucial that the compressed pictures to keep the major details needed for diagnosis. On the basis of this statement, we should utilize the maximum amount a priori info as possible, for instance, what sort of attributes needs to be preserved and which section of the image are essential for diagnosis.

PCA is usually a mathematical formulation employed in the reduction of data dimensions. Thus, the PCA technique permits the identification of standards in information and their expression in such how that their similarities and differences happen to be emphasized, i.e., their dimensions can be reduced while not a lot of loss of information. In summary, the PCA formulation can be utilized as a digital image compression algorithm with a minimum degree of loss [8]. Use of the PCA technique in data dimensionality reduction is justified by easy representation of multidimensional data, using the information contained in the data covariance matrix, principles of linear algebra and basic statistics. The PCA is an authentic image compression algorithm with minimal loss of information [8].

The SPIHT algorithm was developed by Said and Pearlman in 1996. The SPIHT uses the fundamental idea of zero-tree coding from the EZW but is able to obtain a more efficient and better compression performance in most cases without having to use an arithmetic encoder. It uses wavelet sub band decomposition and imposes a quad tree structure across the sub bands in order to exploit the inter-band correlation.

The SPIHT method is not a simple extension of traditional methods for image compression, and represents an important advance in the field. The SPIHT (set partitioning in hierarchical trees) is an efficient image coding method using the wavelet transform. Recently, image coding using the wavelet transform has attracted great attention. Among the many coding algorithms, the embedded zero tree wavelet coding by Shapiro and its improved version, the set partitioning in hierarchical trees (SPIHT) by Said and Pearlman have been very successful. Compared with JPEG the current standard for still image compression, the EZW and the SPIHT are more efficient and reduce the blocking artifact. The method deserves special attention because it provides the following: (a) Good image quality, high PSNR, especially for color images; (b) It is optimized for progressive image transmission; (c) Produces a fully embedded coded file; (d) Simple quantization algorithm; (e) Can be used for lossless compression; (f) Can code to exact bit rate or distortion; (g) Fast coding/decoding (nearly symmetric); (h) Has wide applications, completely adaptive; (i) Efficient combination with error protection. Note that different compression methods were developed specifically to achieve at least one of those objectives. What makes SPIHT really outstanding is that it yields all those qualities simultaneously.

II. Material and Method

A. The Standard PCA Algorithm

The description of the PCA algorithm for compression of picture has been made broadly in [4] and [5] however few variances have been found out within the papers. Since solely grey picture is concerned in our work, the input picture *q (x, y)* or *Z* is of associate degree M×N monochrome picture where by every element signify the intensity value.

$$q(x,y)=\begin{bmatrix} q(0,0) & q(0,1) & \cdots & q(0,N-1) \\ q(1,0) & q(1,1) & \cdots & q(1,N-1) \\ \vdots & \vdots & \vdots & \vdots \\ q(M-1,0) & q(M-1,1) & \vdots & q(M-1,N-1) \end{bmatrix}_{(M\times N)}$$

Step I: Subtract the mean

PCA is begun by producing information set whose mean is zero. As a consequence, all elements are going to be redressed by the expedient of the mean on the row, leading to row element adjust, $\overline{q(x,y)}$or $\overline{Z}$.

Step II: Calculation of covariance matrix, then eigenvalues and eigenvectors of the covariance matrix.

Covariance measures the linear relationship between variables and corresponding matrix will be a square matrix in the type of:

$$Co=\begin{bmatrix} cov(Z_1,Z_1) & cov(Z_1,Z_2) & \cdots & cov(Z_1,Z_k) \\ cov(Z_2,Z_1) & cov(Z_2,Z_2) & \cdots & cov(Z_2,Z_k) \\ \vdots & \vdots & \vdots & \vdots \\ cov(Z_k,Z_1) & cov(Z_k,Z_2) & \vdots & cov(Z_k,Z_k) \end{bmatrix}$$

The off-diagonal elements $Co\,(i,j)$ signify the covariance's of columns i and j. The diagonal elements $Co\,(i,j)$ signify the variances for the columns of $q(x,y)$.

Step III: Culling elements and framing feature vector

The eigenvalues and eigenvectors of the matrix obtained are calculated. In fact, it turns out that the eigenvector with the highest eigenvalue is the principal component of the picture. As soon as the quantity of primary components for eigenvectors is resolute, a feature vector can be framed containing all the culled eigenvectors, λ.

$$Feature\_Vector = FV = [\lambda_1,\lambda_2,\lambda_3 \dots \lambda_k]_{(N\times k)}$$

Step IV: Deriving the final compressed picture

$$Final\_data = FD = \left[FV^T \times \overline{q(x,y)}^T\right]_{(k\times M)}$$

As of right now, the *Final_data* acquired has diminished dimensionality in which data items are organized in columns and dimensions along rows. In picture compression, it's of enthusiasm to figure out how the original picture may appear with the *Final_data*. The level of picture debasement and data compression can rely upon the quantity of principal components chose. The lower the amount of principal components, the more the information will be compacted to the detriment of picture quality. To recover back the original picture, with or with no whole set of eigenvectors,

$$PCA\_Image = (FV * FD)^T + mean$$

In this work, compression ratio relation is developed on the basis of the matrix size saved in the compressed info. The amount of principal component used, $PC \text{ or } k$ will depend on the compression level desired:

$$CR = 1 - \frac{k}{M}$$

Where *M* signify the number of row of the input picture.

B. Block-by-block PCA

Rather than compressing the entire picture at once, we tend to have an interest in acting on the sub - block of the original picture. Block-by-block PCA was initially proposed by Taur and Tao [21] after they examined the thought of implementing a distinct amount of primary components for every blocks in the picture. Input image is partitioned-off into blocks of dimension n and PCA algorithm was implemented one-by-one on every blocks. Each block $Z_{i\text{th}}$ consists of intensity values $q(x,y)$ where i^{th} signify the block number of the picture.

$$Z_{i^{th}} = \begin{bmatrix} q(0,0) & q(0,1) & \cdots & q(0,n-1) \\ q(1,0) & q(1,1) & \cdots & q(1,n-1) \\ \vdots & \vdots & \vdots & \vdots \\ q(n-1,0) & q(n-1,1) & \vdots & q(n-1,n-1) \end{bmatrix}_{(n\times n)}$$

In this work, compression ratio relation is developed on the basis of the matrix size saved in the compressed info:

$$\mathrm{CR_{b/b}} = 1 - \left(\frac{\mathrm{n \times k}}{\mathrm{M \times N}} \times \frac{\mathrm{M \times N}}{\mathrm{n^2}}\right)$$

Where $\frac{\mathrm{M\times N}}{\mathrm{n^2}}$ is the total number of blocks for a picture with dimension M×N.

C. Block-to-row PCA

Like block-by-block PCA, the original picture is divided into n×n blocks. As in [15], every block is concatenated into row to get a remodeled matrix,

$$\mathrm{D} = [\mathrm{z_1, z_2, z_3} \ldots \mathrm{z_{block}}]_{(\mathrm{n^2 \times block})}$$

Where $\mathrm{z_i}$ comprises all elements within a block,

$$\mathrm{z_{ith}} = \begin{bmatrix} \mathrm{q(0,0)} \\ \mathrm{q(1,0)} \\ \vdots \\ \vdots \\ \mathrm{q(n^2-1,0)} \end{bmatrix}_{(\mathrm{n^2 \times 1})}$$

PCA is subsequently employed on the remodeled matrix. The compressed information acquired during this case is,

$$\mathrm{FD} = \left[\mathrm{FV^T} * \overline{\mathrm{D}}^{\mathrm{T}}\right]_{(\mathrm{n^2 \times k})}$$

Where FV is the feature vector and $\overline{\mathrm{D}}$ the mean-adjusted matrix. This compressed image can be reconstructed utilizing backwards strategy that has been utilized in concatenation. Consequently, the compression ratio, again on basis of the compressed information, is characterized as:

$$\mathrm{CR_{b2r}} = 1 - \left(\frac{\mathrm{n^2 \times k}}{\mathrm{M \times N}}\right)$$

D. SPIHT

The SPIHT algorithm was developed by Said and Pearlman in 1996. The SPIHT uses the fundamental idea of zero-tree coding from the EZW but is able to obtain a more efficient and better compression performance in most cases without having to use an arithmetic encoder. It uses wavelet sub band decomposition and imposes a quad tree structure across the sub bands in order to exploit the inter-band correlation.

SPIHT algorithm uses a special data structure – spatial orientation trees (SOT). This particular structure is not only made full use of different scales the correlation between the wavelet coefficients, but also give full consideration to the correlation of the same scale wavelet coefficients. Fig 1 shows the tree structure used in SPIHT.

The algorithm searches each tree, and partitions the tree into one of three lists: (a) the list of significant pixels (LSP) containing the coordinates of pixels found to be significant at the current threshold; (b) the list of insignificant pixels (LIP), with pixels that are not significant at the current threshold; and (c) the list of insignificant sets (LIS), which contain information about trees that have all the constituent entries to be insignificant at the current threshold. [22-23]

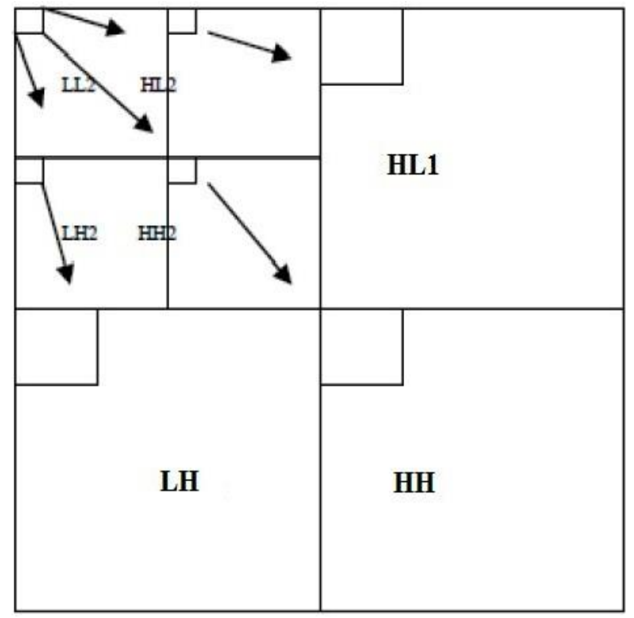


Fig 1 Tree structure used in SPIHT algorithm

The SPIHT algorithm consists of three stages: initialization, sorting pass and refinement pass. At the initialization stage the SPIHT first defines a start threshold according to the maximum value in the wavelet coefficients pyramid, then sets the LSP as an empty list and puts the coordinates of all coefficients in the coarsest level of the wavelet pyramid (LL band) in the LIP and those which have descendants to the LIS.

In the sorting pass, the elements in the LIP then in the LIS are sorted. For each pixel in the LIP it performs a significance test against the current threshold and outputs the test result (0 or 1) to the output bit stream. If a coefficient is significant, its sign is coded and then its coordinate is moved to the LSP. During the sorting pass of LIS, the SPIHT does the significance test for each set in the LIS and outputs the significance information (0 or 1). If a set is significant, it is partitioned into its offspring and leaves. The current threshold is divided by 2 and the sorting and refinement stages are continued until we achieve the target bit-rate. [24]

Steps of SPIHT Algorithm as follows:

1. Initialize the LIP, LIS, LSP table and determine the maximum threshold

   Threshold $T=2^n$,

   Where, n= [lg2max| (i, j) |]

2. For a given threshold, searches LIP, LIS table to determine the importance of each wavelet coefficient in LIP table.
3. If coefficient (i,j) is significant, then the output "1" and the sign bit is sent out , the node removed from the LIP form, added to the LSP end of the table.
4. If coefficient (i,j) is not significant, then the output should be "0", do not remove this node, the corresponding

coordinates are moved to the LIP or LIS respectively, for subsequent testing at a lower bit level.

5. For the same threshold value, scan each node fine in turn in LSP table fine: output not newly added node LSP table wavelet coefficients corresponding to the first binary representation of n +1 bits, the scan end.

6. For the next scan: threshold T←T/2, n←n-1, repeat step (3) and step(4) step(5), until the threshold values or bit rate compliance encoder requirements.

In this work, compression ratio relation is developed on the basis of the matrix size saved in the compressed info:

$$CR_{SPIHT} = \frac{Size\ of\ Uncompressed\ image}{Size\ of\ Compressed\ image}$$

This value should be greater than one for good compression, its value one signifies no compression takes place. The compression ratio in form of percentage (Space Saving) can be expressed as:

$$CR_{\%} = \left(1 - \frac{Size\ of\ Compressed\ image}{Size\ of\ Uncompressed\ image}\right) \times 100$$

E. PERFORMANCE PARAMETER

The *PSNR* is the peak signal-to-noise ratio, in decibels, between two images. This ratio is often used as a quality measurement between the original and a compressed image. The higher the PSNR, the better the quality of the compressed image. The *Mean Square Error (MSE)* and the *Peak Signal to Noise Ratio (PSNR)* are the two error metrics used to compare image compression quality. The MSE represents the cumulative squared error between the compressed and the original image, whereas PSNR represents a measure of the peak error. The lower the value of MSE, the lower the error,

$$PSNR = 20 \times log_{10}\left(\frac{R}{\sqrt{MSE}}\right)$$

Where the MSE (Mean Square Error),

$$MSE = \frac{1}{M \times N}\sum_{0}^{M-1}\sum_{0}^{N-1}\|g(i,j) - p(i,j)\|^2$$

g - Original picture Pixel Value

p - Compressed picture Pixel Value

M - Number of rows of the picture and i signifies the index of the row

N - Number of columns of the picture and j signifies the index of the column.

R - Maximum Fluctuation in the image data type. For example, if the input image has a double-precision floating-point data type, then R is 1. If it has an 8-bit unsigned integer data type, R is 255, etc.

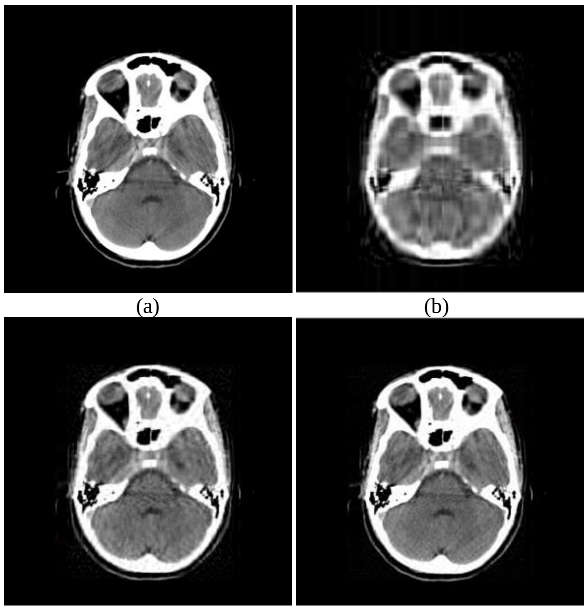
The *SSIM (Structural Similarity Index Matrix)* is a method for predicting the perceived quality of digital images and videos. SSIM is used for measuring the similarity between two images. The SSIM index is a full reference metric; in other words, the measurement or prediction of image quality is based on an initial uncompressed or distortion – free image as reference. SSIM is designed to improve on traditional methods such as PSNR and MSE, which have proven to be inconsistent with human visual perception [25]. The PSNR and MSE approach estimate absolute errors; on the other hand, SSIM is a perception-based model that considers image degradation as perceived change in structural information, while also incorporating important perceptual phenomena, including both luminance masking and contrast masking terms. Luminance masking is a phenomenon whereby image distortions tend to be less visible in bright regions, while contrast masking is a phenomenon whereby distortions become less visible where there is significant activity or texture in the image. The SSIM index is calculated on various windows of an image. The measure between two windows x and y of common size N×N is:

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + C1)(2\sigma_{xy} + C2)}{(\mu_x^2 + \mu_y^2 + C1)(\sigma_x^2 + \sigma_y^2 + C2)}$$

Where μ_x the average of x; μ_y the average of y; σ_x^2 the variance of x; σ_y^2 the variance of y; σ_{xy} the covariance of x and y; $C1 = (k_1L)^2$ & $C2 = (k_2L)^2$ two variables to stabilize the division with weak denominator; L the dynamic range of pixel-values (typically $2^{\#bits\ per\ pixel} - 1$); and k_1=0.01 and k_2=0.03 by default.

III. RESULTS AND DISCUSSION

In this section, the chosen medical image is tested on a Intel (R) core 2 duo 2.40 GHz Computer utilizing MATLAB 7.8.0. The input original picture is initially converted to gray scale and re-sized to 512 × 512. In original image, we first apply standard PCA algorithm. The Fig 2 shows the compressed image with different number of principal component. The Table I also shows the values of different parameter for different number of principal component.



(a) (b) (c) (d)

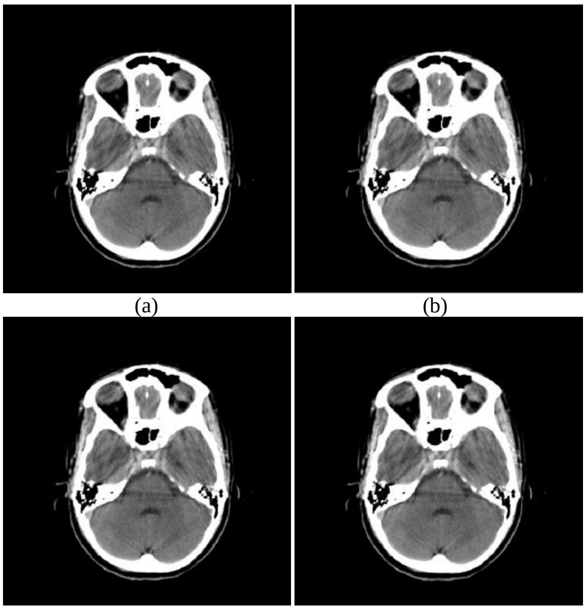
Fig 2 Standard PCA method (a) Original image; (b) Compressed image PC=10/360, ssim=0.74; (c) Compressed image PC=30/360, ssim=0.83; (d) Compressed image PC=50/360, ssim=0.88.

TABLE I. STANDARD PCA

Standard PCA		Total PC = 360	
PCs	**PSNR (dB)**	**CR**	**SSIM**
10	33.1711	0.98047	0.74027
20	34.4732	0.96094	0.79818
30	35.3986	0.94141	0.83501
40	36.4544	0.92188	0.86412
50	37.6351	0.90234	0.88601
60	38.6788	0.88281	0.90464
70	39.6767	0.86328	0.91957
80	40.6767	0.84375	0.93315
90	41.6248	0.82422	0.94341
100	42.544	0.80469	0.95247
110	43.5399	0.78516	0.96047
120	44.4936	0.76563	0.96722
130	45.4915	0.74609	0.9733
140	46.4931	0.72656	0.97816
150	47.508	0.70703	0.98236

From the Table I it is clear that if we increase the number of principal component, the quality of image improves and at the same time compression of image reduces.

Now in original image, we apply the first block based method i.e. Block by block PCA. In block by block method, original image is partitioned-off into blocks of dimension n and PCA algorithm was implemented one-by-one on every blocks. The Fig 3 shows the compressed image with different number of principal component applied on each block. The Table II and III also show the values of different parameter for different number of principal component for different block size.



(a) (b) (c) (d)

Fig 3 Block by Block PCA (Block size=8) (a) Original image; (b) Compressed image PC=2/8, ssim=0.992; (c) Compressed image PC=4/8, ssim=0.999; (d) Compressed image PC=6/8, ssim=1.

The Table II and III show the objective evaluation result for block-by-block PCA in block size = 8 and 16. It is clear from the table II and III that if we increase the block size, the quality of image decreases and computation time reduces for same value of compression ratio. Simply we can say image quality improves with the reduction of compression ratio and the reduction of block size.

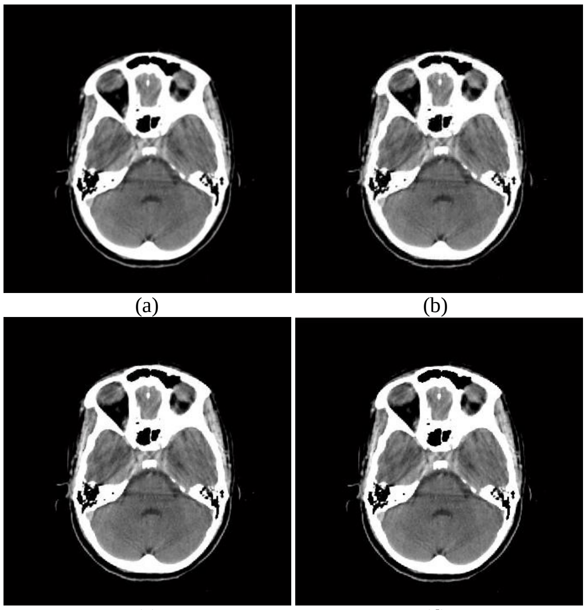
TABLE II. BLOCK-BY-BLOCK PCA FOR BLOCK SIZE = 8

Block size = 8		Total PC = 8	
PCs	**PSNR (dB)**	**CR**	**SSIM**
PC = 1	40.6905	0.875	0.9734
PC = 2	45.0583	0.75	0.9928
PC = 3	49.5028	0.625	0.998
PC = 4	54.6314	0.5	0.9994
PC = 5	61.0348	0.375	0.9999
PC = 6	71.3159	0.25	1

TABLE III. BLOCK-BY-BLOCK PCA FOR BLOCK SIZE = 16

Block Size = 16		Total PC = 16	
PCs	**PSNR (dB)**	**CR**	**SSIM**
PC = 1	37.0464	0.9375	0.9139
PC = 2	39.512	0.875	0.9598
PC = 3	41.7967	0.8125	0.979
PC = 4	43.9064	0.75	0.9883
PC = 5	46.1636	0.6875	0.9939
PC = 6	48.4557	0.625	0.9965
PC = 7	50.7795	0.5625	0.9981
PC = 8	53.1744	0.5	0.9989
PC = 9	55.9243	0.4375	0.9994
PC = 10	58.9374	0.375	0.9997
PC = 11	62.8267	0.3125	0.9999
PC = 12	67.0852	0.25	0.9999

Now in original image, we apply the second block based method i.e. Block to row PCA. In block to row method, original image is divided into n×n blocks and every block is concatenated into row to get a remodeled matrix. The Standard PCA is applied on remodeled matrix to compress the image. The Fig 4 shows the compressed image with different number of principal component applied on remodeled matrix.



(a) (b) (c) (d)

Fig 4 Block to Row PCA (Block size=8) (a) Original image; (b) Compressed image PC=10/63, ssim=0.960; (c) Compressed image PC=30/63, ssim=0.990; (d) Compressed image PC=50/63, ssim=0.997.

The Table IV and V also show the values of different parameter for different number of principal component for different block size.

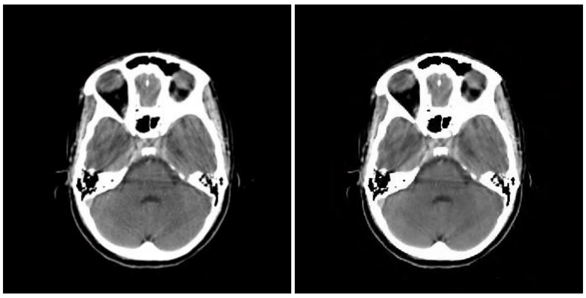
TABLE IV. BLOCK-TO-ROW PCA FOR BLOCK SIZE = 8

Block size = 8		Total PC =63	
PCs	**PSNR (dB)**	**CR**	**SSIM**
10	39.324	0.99756	0.96074
20	41.9855	0.99512	0.98315
30	43.9848	0.99268	0.99054
40	46.3361	0.99023	0.99467
50	49.0942	0.98779	0.99784
60	58.1698	0.98535	0.99961

TABLE V. BLOCK-TO-ROW PCA FOR BLOCK SIZE = 16

Block Size = 16		Total PC = 255	
PCs	**PSNR (dB)**	**CR**	**SSIM**
10	35.744	0.99023	0.8677
20	37.1771	0.98047	0.91324
30	38.2226	0.9707	0.93591
40	39.0802	0.96094	0.94776
50	40.0517	0.95117	0.958
60	41.0751	0.94141	0.96616
70	42.12	0.93164	0.97221
80	43.2908	0.92188	0.97777
90	44.4506	0.91211	0.98177
100	45.6607	0.90234	0.9856
110	46.8997	0.89258	0.98863
120	48.1406	0.88281	0.99082
130	49.407	0.87305	0.99269
140	50.7046	0.86328	0.99424
150	51.995	0.85352	0.99551

Now in original image, we apply the SPIHT method i.e. wavelet method. To choose SPIHT method because it gives good reconstruction quality of image as well as good compression ratio. The Fig 5 shows the compressed image when original image is compressed by SPIHT. Here original image is considered as img1.



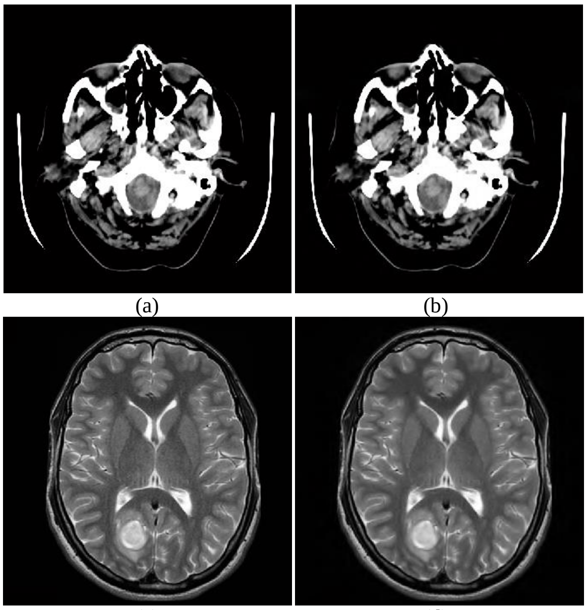
(a) (b)

Fig 5 SPIHT Method (a) Original image (img1); (b) Compressed image, ssim=0.952.

TABLE VI. SPIHT METHOD APPLIES ON DIFFERENT IMAGE

SPIHT			
Image	**PSNR**	**CR (Space Saving)**	**SSIM**
img1	40.7004	0.78453	0.95289
img2	44.4541	0.89923	0.97355
img3	39.2634	0.782	0.94003

PSNR (Peak signal-to-noise ratio) in dB is employed so as to review the grade of the reconstructed image changes with compression ratio. PSNR value equals to infinity if there isn't any quality disparity involving the original image as well as in the reconstructed image. Compression ratio equals to zero if all primary components are culled while compression ratios proceed towards to one if lesser primary components are utilized.



(a) (b) (c) (d)

Fig 6 SPIHT method (a) Original image (img2); (b) Compressed image, ssim=0.973; (c) Original image (img3); (d) Compressed image, ssim=0.940.

Here we perform on image size of 512 × 512, so the number of principal component selected for maximum compression is approximately 2 % of total number of column of original image. In ROI region we take the principal component according to the size of block used. In block-by-block PCA, Each block is consider for compression, so according to the block size used we have selected the principal component is approximately 25%, 50% and 75% of the block size used. Finally in Block-to-row PCA, we select the principal component on the basis of the total number of block concatenated to form a row. Table I and II demonstrate the objective assessment results for block-by-block PCA algorithm in ROI region using block size of 8 and 16 respectively. Table III and IV present the objective assessment result for block-to-row PCA algorithm in ROI region using block size of 8 and 16 respectively.

Fig 3 is an original gray image, by using this image we first done the segmentation. After extracting ROI we have applied Block based PCA algorithm on ROI region. In Non-ROI region we have applied general PCA algorithm. Fig 4 shows the compressed image having ROI compression using block-by-block PCA and non-ROI compression using general PCA. The overall compression ratio is about 89%.

Fig 5 shows the compresses image having ROI compression using Block-to-row PCA and non-ROI compression using general PCA. The overall compression ratio is about 98%.

As obvious from fig 4 & 5, the PSNR as well as the compression ratio are higher in block to row PCA algorithm as in comparison with in block through block PCA algorithm.

IV. Conclusion

On this work we've executed an effort to judge and compare 2 extended PCA algorithms on Region of Interest of a medical image. From this work, it's truly concluded that the compression operation of block-to-row PCA excel the block-by-block PCA, both in terms of PSNR, visual inspection and compression ratio. This will give maximum compression of ROI region with minimal loss of image-quality. The non-ROI region was compressed by general PCA. This gives very high compression but at the same time loss of the image quality. Hence, using region separation technique allows a compression ratio of up to 98% with no loss of information on ROI region. Future work incorporates localizing the ROI in a more precise manner for instance to segment the precise shape of ROI outlining the border since almost all of the clinically vital data is in arbitrary structure.

References

[1] Lim Sin Ting, David Yap Fook Weng and Nurulfajar Bin Abdul Manap, “A Novel Approach for Arbitrary-Shape ROI Compression of Medical Image Using Principal Component Analysis (PCA)”, Trends in Applied Sciences Research, pp. 68-76, Academic Journals Inc, 2015.

[2] Amit S. Tajne, “A Survey on medical image compression Using Hybrid Technique”, International Journal of computer science and Mobile Computing, Vol. 4, issue 2, February 2015, pg. 18-23.

[3] Prabhdip Kaur, “Hybrid PCA-DCT Based image Fusion for Medical Images”, International Journal of Technology enhancement and emerging Engineering Research, Vol 3, Issue 7, 2015.

[4] S. T. Lim, D. F. W. Yap and N. A. Manap, “Medical image compression Using Block-Based PCA algorithm”, IEEE International Conference on Computer, Communication, and control technology (I4CT 2014), pp. 171-175, Langkawi, kedah, Malaysia, 2014.

[5] S. T. Lim, D. F. W. Yap and N. A. Manap, “A GUI System for Region-based image compression Using Principal Component Analysis”, IEEE International Conference on Computer, Communication, and control technology (I4CT 2014), 2014.

[6] Nitu Rani, Savita Bishnoi, "Comperative Analysis of Image Compression using DCT and DWT Transforms", International Journal of computer Science and Mobile Computing, Vol. 3, issue 7, July 2014.

[7] K. Gopi, Dr. T. Rama shri, "Medical Image Compression using Wavelets", IOSR Journal of VLSI and signal processing, vol. 2, issue 4, june 2013.

[8] R. E. Santo, “Principal Component Analysis applied to digital image compression”, Einstein (São Paulo), vol. 10, pp. 135-139, 2012.

[9] M. Mofarreh-Bonab and M. Mofarreh-Bonab, “A new technique for image compression using PCA”, International Journal of Computer Science & Communication Networks, vol. 2, pp. 111-116, 2012.

[10] G. M. Padmaja, P. Nirupama, "Analysis of various image compression Techniques", ARPN Journal of science and Technology, vol 2, No 4, may 2012.

[11] S. Stolevski, “Hybrid PCA algorithm for image compression”, 18th Telecommunication forum TELFOR 2010, pp. 685-687, 2010: Serbia, Belgrade.

[12] J. Shlens, “A tutorial on Principal Component Analysis”, 2009.

[13] G. Reishofer, S. Ropele and R. Stollberger, “Optimized Data Compression and denoising with PCA using a kurtosis Based selection Criterion”, Proc. Intl. Soc. Mag. Reson. Med. 15, 2007.

[14] Liminita STATE, Catalina Lucia COCIANU and Vlamos PANAYIOTIS, “Neural Network for Principal Component analysis with applications in image compression”, Systemics, Cybernetics and Informatics, Vol 5 No. 2, 2007.

[15] A. Dwivedi, A. Tolambiya, P. Kandula, N.S.C. Bose, A. Kumar, P.K. Kalra, “Color image compression using 2-Dimensional Principal Component Analysis (2DPCA)”, in Proc. Of ASID, pp. 488-491, 2006, New Delhi.

[16] Se-Kee Kil, Jong-Shill Lee, Dong-fan Shen, Je-Goon Ryu, Eung-Hyuk Lee, Hong-Ki Min and Seung-Hong hong, “Lossless Medical image compression using Redundancy Analysis”, IJCSNS International Journal of Computer Science and Network Security, Vol 6 No. 1A, January 2006.

[17] Anke Meyer-Base, Karsten Jancke, Axel Wismuller, Simon Foo and Thomas Martinetz, “Medical image compression using topology-preserving neural networks”, Engineering Applications of Artificial Intelligence, pp. 383-392, 2005.

[18] Jianxun MI and De-Shuang Huang, “Image compression using Prncipal component Neural Network”, 8th international conference on control, automation, Robotics and Vision kunming, China, 6-9th December 2004.

[19] Costa S. And S. Fiori, “Image compression using Principal component neural networks”, Image vission and computing, 2001.

[20] Salih Burak Gokturk, Carlo Tomasi, Bernd Girod and Chris Beaulieu, “Medical Image Compression Based on region of interest with application to Colon CT images”, Engineering in Medicine and Biology Society, 2001. Proceedings of the 23rd Annual International Conference of the IEEE (Volume: 3), 2001.

[21] J. S. Taur and C.W. Tao, “Medical image compression using principal component analysis”, IEEE Internation Conference on Image Processing, vol. 2, pp. 903-906,1996.

[22] Sadashivappa, G., Mahesh Jayakar, and KVS Ananda Babu. "Analysis of SPIHT Algorithm Using Tiling Operations." Signal Acquisition and Processing, 2010. ICSAP'10. International Conference on. IEEE, 2010.

[23] Jiang, Chunlei, and Shuxin Yin. "A Hybrid Image Compression Algorithm Based on Human Visual System." *2010 International Conference on Computer Application and System Modeling (ICCASM 2010)*. Vol. 9. IEEE, 2010.

[24] Zhu, Li, and Yi-min Yang. "Embedded Image Compression Using Differential Coding and Optimization Method." *Wireless Communications, Networking and Mobile Computing (WiCOM), 2011 7th International Conference on*. IEEE, 2011.

[25] Wang, Zhou, et al. "Image quality assessment: from error visibility to structural similarity." *IEEE transactions on image processing* 13.4 (2004): 600-612.